$2^η$ Σειρά Γραπτών Ασκήσεων

Αλγόριθμοι και Πολυπλοκότητα

Σεραφείμ Τζελέπης ΑΜ:el18849

Άσκηση 1)

Ξεκινώντας τον αλγόριθμο μας πρέπει να βρούμε για κάθε σημείο k, δύο σημεία τα οποία του αντιστοιχούν έστω λ_k και ρ_k τα οποία βρίσκονται πάνω στην ευθεία l και το λ_k είναι το αριστερότερο εκ των δύο. Τα σημεία λ_k και ρ_k είναι τέτοια ώστε d(k, λ_k) = d(k, ρ_k) = r. Η εύρεση αυτών των σημείων για κάθε k είναι σταθερού χρόνου. Συνεπώς έχουμε πλέον όλα τα 2n(μπορεί κάποια να ταυτίζονται σε περίπτωση που το αρχικό σημείο βρίσκεται σε απόσταση r από την ευθεία) σημεία τα όποια τα ταξινομούμε βάση το δεξί τους άκρο πάνω στην ευθεία. Επομένως έχοντας κάνει αυτήν την προεργασία έχουμε τα διάφορα λ_1, λ_2,…,λ_n και ρ_1,ρ_2,…,ρ_n πάνω στην ευθεία μας, χωρίς απαραίτητα αυτήν την σειρά. Το μόνο που γνωρίζουμε σίγουρα είναι ότι λ_k < ρ_k για κάθε k. Ξεκινάμε πλέον την εύρεση του ελάχιστου αριθμού δίσκων χρησιμοποιώντας δύο αρχικώς άδεια σύνολα seen={} και visited={}, και διατρέχουμε την ευθεία από τα αριστερά προς τα δεξιά ξεκινώντας από το αριστερότερο άκρο.

- Αν συναντήσω κάποιο στοιχείο λ_k το βάζω στο σύνολο seen το k. Και συνεχίζω να διατρέχω την ευθεία προς τα δεξιά.

- Αν συναντήσω κάποιο στοιχείο ρ_k τότε ελέγχω αν υπάρχει το k στο σύνολο visited.

    - Αν υπάρχει τότε συνεχίζω να διατρέχω την ευθεία προς τα δεξιά
    - Αν δεν υπάρχει τότε βάζω όλα τα στοιχεία του seen στο σύνολο visited καθώς ένας κύκλος με κέντρο το ρ_k θα τα περιέχει όλα τα στοιχεία του seen. Επίσης αυξάνουμε τον μετρητή των κύκλων κατά 1 και θέτουμε το seen ξανά ως empty set. Ελέγχουμε αν το επαυξημένο σύνολο visited περιέχει όλα τα σημεία n του προβλήματος.

- Αν ναι, τελείωσε ο αλγόριθμος μας.
- Αν όχι, συνεχίζουμε διατρέχοντας στο επόμενο στοιχείο προς τα δεξιά.

Συνεπώς τελικά διατρέχουμε μία φόρα αυτά τα 2n σημεία και επίσης χρειάζεται να κάνουμε μια ταξινόμηση συνεπώς η χρονική πολυπλοκότητα της λύσης μας είναι O(nlogn) λόγω της ταξινόμησης.

Απόδειξη ορθότητας:

Έστω $o_1,o_2,\ldots,o_\pi$ τα κέντρα αυγό βραστό των κύκλων της βέλτιστης λύσης και έστω $s_1,s_2,\ldots,s_m$ τα κέντρα των κύκλων της λύσης μας.

Αρχικά θα αποδείξουμε ότι για κάθε r <= m ισχύει ότι $s_r >= o_r$ με την μέθοδο της επαγωγής.

Για r = 1 ισχύει καθώς ξέρουμε ότι βάσει του αλγόριθμου μας το s_1 θα βρίσκεται στο ρ_1 άρα στο δεξιότερο πιθανό κέντρο ενός κύκλου που περιέχει το πρώτο στοιχείο συνεπώς $o_1 <= s_1$.

Έστω ότι ισχύει για $o_{r-1} <= s_{r-1}$ τότε θα αποδείξω ότι ισχύει $o_r <= s_r$. Έστω f_i το επόμενο διάστημα με λ_i > s_{r-1} τότε ο άπληστος αλγόριθμος θα δημιουργήσει ένα νέο σημείο το όποιο s_r = ρ_i όμως το o_{r-1} πρέπει να καλύψει και αυτό το διάστημα f_i τότε θα ισχύει $o_r <= s_r$.

Έστω k > m. Αν ο αλγόριθμος χρειάζεται να προσθέσει κάποιο ακόμη σημείο μετά το s_m, αυτό σημαίνει ότι υπάρχει ένα διάστημα f_j με ρ_j > s_m.

Δεδομένου ότι $o_m \leq s_m$, o_m < ρ_j . Έτσι o_m δεν καλύπτει το f_j . Όμως το o_m είναι το πιο δεξί σημείο

της βέλτιστης λύσης που αντιτίθεται στην αποδοχή της o1, o2, . . . , o_m ως εφικτής λύσης.

Άσκηση 2)

A)

Αρχικά για την επίλυση του ερωτήματος αυτού υπολογίζουμε τον λόγο $r_i = w_i/p_i$

για i = 0,1,…,n. Για να ελαχιστοποιήσουμε τον βεβαρυμμένος χρόνο εξυπηρέτησης θέλουμε να εξυπηρετούμε πρώτα αυτούς με την μεγαλύτερη σημασία, και θέλουμε επίσης να εξυπηρετούμε πρώτα αυτούς με την ελάχιστο χρόνο. Συνεπώς αυτά τα δυο κριτήρια περιγράφονται από τον λόγο r που ορίσαμε παραπάνω, αποδεικνύεται συνεπώς ότι η σειρά εξυπηρέτησης θα είναι βάση αυτού του λόγου. Συνεπώς ταξινομούμε τα r. Επομένως ο πρώτος πελάτης που θα εξυπηρετηθεί θα είναι αυτός με τον μεγαλύτερο λόγο, ο δεύτερος που θα εξυπηρετηθεί θα είναι αυτός με τον δεύτερο μεγαλύτερο λόγο κλπ. Συνεπώς ο ελάχιστος βεβαρυμμένος χρόνος εξυπηρέτησης θα είναι $w_{max}p_{max} + w_{max-1}(p_{max-1} + p_{max}) + \ldots + w_{min}(p_{min} + \ldots + p_{max})$, όπου τα στοιχεία με δείκτη max είναι ουσιαστικά αυτά που αντιστοιχούν στον μέγιστο λόγο , και τα στοιχεία με δείκτη min είναι αυτά που αντιστοιχούν στον ελάχιστο λόγο.

Β)

Για το δεύτερο ερώτημα θα χρησιμοποιήσουμε την προσέγγιση του δυναμικού προγραμματισμού. Ουσιαστικά κάθε φόρα θα παίρνουμε το min του να βάλουμε τον πελάτη στον πωλητή 1 ή να μην τον βάλουμε, δηλαδή να τον βάλουμε στον πωλητή 2.

Άσκηση 3)

Α)

Για την λύση μας θα ξεκινήσουμε από το τελευταίο σημείο x_f και θα δούμε ποιο σημείο x_i είναι βέλτιστο ώστε να χρησιμοποιηθεί ως η αρχή της τελευταίας στέγης. Συνεπώς ο αναδρομικός τύπος που θα χρησιμοποιήσουμε θα είναι:

$cost(i) = \min_{0<=j<=i}\{cost(j – 1) + (x_i – x_j)^2 + C\}$,

Με cost(i) το ελάχιστο κόστος για να καλύψουμε τα σημεία $x_0 = 0$ έως x_i με cost(-1) = 0. Για τον υπολογισμό του κάθε cost έχουμε πολυπλοκότητα n, συνεπώς τελικά θα κάνουμε n υπολογισμούς πολυπλοκότητας n επομένως η συνολική πολυπλοκότητα είναι $\Theta(n^2)$.

B)

Ουσιαστικά θέλουμε να βρούμε μια διάταξη των ευθείων i(1),i(2),..i(m) με τρόπον τέτοιο ώστε η i(1) να έχει ελάχιστη τιμή από το -άπειρο ως το σημείο τομής με την i(2), η i(2) να έχει ελάχιστη τιμή από το σημείο τομής της με την i(1) μέχρι το σημείο τομής της με την i(3) και αντιστοίχως και για τις επόμενες ευθείες. Οι κλίσεις που μας δίνονται για τις ευθείες είναι ταξινομημένες και κάθε ευθεία που εισάγεται στην δομή μας, πιθανώς να διαγράφει κάποιες από τις προηγούμενες. Συνολικά όμως κάθε ευθεία θα εισαχθεί ή θα διαγραφθεί από την δομή μας το πολύ μία φόρα, και επειδή υπάρχουν συνολικά n ευθείες μπορούμε να σχηματίσουμε αυτή τη δομή σε χρόνο Θ(n). Έχοντας έτοιμη την δομή, για κάθε σημείο x_i ξεκινάμε από τη πρώτη ευθεία μέχρι να βρούμε αυτήν η οποία καλύπτει το x_i την οποία και επιστρέφουμε. Όμως και τα σημεία είναι ταξινομημένα συνεπώς ξέρουμε ότι η ευθεία η οποία καλύπτει το σημείο x_{i+1} θα είναι ή η ευθεία που καλύπτει και το x_i ή αλλιώς θα βρίσκεται πιο μετά στην δομή μας, έτσι δεν χρειάζεται να ξεκινήσουμε από την αρχή αλλά από την τελευταία ευθεία που επιστρέψαμε συνεπώς θα διασχίσουμε μια φόρα την δομή μας με τις ευθείες με συνολική πολυπλοκότητα Θ(n + k).

Η παρακάτω αναδρομική σχέση μπορεί να ταυτιστεί με αυτή του πρώτου ερωτήματος:

$$cost(i) = x_i^2 + C + \min_{0<=j<=i}\{ a[j]x_i + b[j]\}$$

Με $a[j] = -2x_j$ και $b[j] = cost(j – 1) + x_j^2$. Η σχέση αυτή μας δείχνει ότι το πρόβλημα μας είναι ισοδύναμο με την εύρεση ελάχιστης τιμής από σύνολο ευθειών με κλίσεις a[j] στα σημεία x_i, και επειδή τόσο οι κλίσεις όσο και τα σημεία είναι ταξινομημένα μπορούμε να υπολογίσουμε όλα τα ελάχιστα σε χρόνο Θ(n), που αποτελεί ξεκάθαρη βελτιστοποίηση από αυτόν του πρώτου ερωτήματος.

Άσκηση 4)

Αυτό που πρέπει να κάνουμε στο συγκεκριμένο πρόβλημα είναι να διαχωρίσουμε N φοιτητές σε k λεωφορεία έτσι ώστε να ελαχιστοποιήσουμε τον συνολικό δείκτη ευαισθησίας. Ο τρόπος που θα προσεγγίσουμε το εξής πρόβλημα είναι με την μέθοδο του δυναμικού προγραμματισμού. Πιο συγκεκριμένα έστω MinSens(i,j) των φοιτητών 1 έως i σε j λεωφορεία τότε η ζητούμενη τιμή είναι η MinSens(n,k). Αν πρέπει να τοποθετήσω n φοιτητές σε n λεωφορεία τότε προφανώς η βέλτιστη λύση είναι να τοποθετήσω έναν φοιτητή σε κάθε λεωφορείο έχοντας μηδενικό δείκτη ευαισθησίας. Αν έχω n φοιτητές με 1 λεωφορείο τότε ο συνολικός δείκτης ευαισθησίας θα είναι $\sum_{1\leq\lambda\leq n}\sum_{\lambda\leq m\leq n} A_{\lambda m}$. Στην περίπτωση που οι φοιτητές είναι περισσότεροι από τα λεωφορεία τότε βασιζόμενη όπως προ είπαμε, σύμφωνα με την προσέγγιση του δυναμικού προγραμματισμού, ξεκινάμε από το τέλος και επιλέγουμε ποιος φοιτητής θα τοποθετηθεί στο τελευταίο λεωφορείο. Συνεπώς θα χρησιμοποιήσουμε την αναδρομική σχέση:

$$\text{MinSens}(i,j) = \min_{k<j<i}(\text{MinSens}(j-1, k-1) + \sum_{j\leq\lambda\leq n}\sum_{\lambda\leq m\leq n} A_{\lambda m})$$

Για τον υπολογισμό του MinSens(n,k) έχουμε συνεπώς n x k υποπροβλήματα. Κάθε υποπρόβλημα υπολογίζει το ελάχιστο σε O(N) ποσότητες και για κάθε μια από αυτές πρέπει να υπολογίσουμε το διπλό άθροισμα της αναδρομής σε $O(N^2)$. Έχουμε έτσι συνολική πολυπλοκότητα για την λύση $O(kN^4)$.

Στην περίπτωση που θέλουμε να διαχωρίσουμε τους φοιτητές με τρόπο ώστε ο μέγιστος δείκτης ευαισθησίας λεωφορείου να είναι ο ελάχιστός θα χρησιμοποιήσουμε παρόμοιο αλγόριθμο. Αρχικά στην περίπτωση που έχουμε ίσα λεωφορεία με τον αριθμό των φοιτητών ή την περίπτωση που έχουμε ένα λεωφορείο ισχύουν όσα αναφέρθηκαν και παραπάνω στο προηγούμενο ζήτημα. Στην περίπτωση που οι φοιτητές είναι περισσότεροι από τα λεωφορεία θα χρησιμοποιήσουμε την εξής αναδρομική σχέση:

$$\text{MinSens}(i,j) = \min_{k<j<i}(\max(\text{MinSens}(j-1, k-1)), \sum_{j\leq\lambda\leq n}\sum_{\lambda\leq m\leq n} A_{\lambda m}).$$

Ουσιαστικά σε αυτήν την περίπτωση δεν ψάχνουμε να βρούμε το άθροισμα του δείκτη ευαισθησίας κάθε λεωφορείου αλλά τους συγκρίνουμε για να βρούμε τον μέγιστο δείκτη από αυτούς και κρατάμε τον ελάχιστο από αυτούς. Η πολυπλοκότητα είναι ίδια με αυτή του προηγούμενου ζητήματος δηλαδή $O(kN^4)$.

Άσκηση 5)

Έχουμε ένα μη κατευθυνόμενο γράφημα G(V, E) με n κορυφές και m ακμές.

A) Έχουμε τα T1, T2 όπου είναι δύο διαφορετικά συνδετικά δέντρα με $e \in T_1 \backslash T_2$ με $T_1 \backslash T_2 \neq \emptyset$. Συνεπώς $T_2 \cup \{e\}$ περιέχει κύκλο C, και υπάρχει ακμή $e' \in C$, $e' \notin T_1$ γιατί αν όλες οι ακμές του C ανήκαν στο T_1 τότε το T_1 θα περιείχε τον κύκλο C. Άρα το $(T_2 \setminus \{e\}) \cup \{e'\}$ δεν περιέχει κύκλο και έχει n – 1 ακμές, είναι συνεπώς ένα συνδετικό δέντρο. Για την εύρεση του e' θεωρούμε T_1, T_2 και e = {a,b} και κάνουμε διάσχιση κατά βάθος από το a στο b στο δέντρο T_2(σε $O(|V|)$) και βρίσκουμε μονοπάτι P που να τα συνδέει. Διασχίζουμε αυτό το μονοπάτι που τα συνδέει και ελέγχουμε κάθε ακμή αν ανήκει στο δέντρο T_1, αν όχι τότε βρέθηκε η ζητούμενη e'.

B) Στο συγκεκριμένο ερώτημα μας ζητείται να αποδείξουμε για T_1, T_2 συνδετικά δέντρα τα οποία ανήκουν στο G με $dH(T_1,T_2)$ το μήκος του συντομότερου μονοπατιού μεταξύ των T_1,T_2 στο H ότι ισχύει $dH(T_1,T_2) = |T_1 \setminus T_2|$. Για να το αποδείξουμε θα χρησιμοποιήσουμε επαγωγική απόδειξη. Εξ ορισμού του H, αν $dH(T_1,T_2) = 1$ τότε $|T_1 \setminus T_2|= 1$. Έστω ότι ισχύει $|T_1 \setminus T_2|= k$ για $dH(T_1,T_2) = k$, τότε θα δείξουμε ότι ισχύει $|T_1 \setminus T_2|= k + 1$ για $dH(T_1,T_2) = k + 1$. Έστω $e \in T_1 \setminus T_2$ τότε λόγω A) θα υπάρχει $e' \in T_2 \setminus T_1$ τέτοιο ώστε $T_1'=(T_1\backslash\{e\})\cup\{e'\} \in H$. Όμως $|T_1'\backslash T_2| = k$ και $dH(T_1, T_2) \leq k + 1$. Από επαγωγική υπόθεση έχουμε ότι αν $dH(T_1, T_2) = k$ τότε $dH(T_1, T_2) = k$. Άρα, αναγκαστικά $|T_1 \setminus T_2| = k + 1$. Επομένως $dH(T_1, T_2) \geq k + 1$ άρα $dH(T_1, T_2) = k + 1$ λόγω της παραπάνω ανισότητας. Εφόσον $dH(T_1, T_2) = k + 1$ υπάρχει $T'_1 \in H$ τέτοιο ώστε $dH(T_1, T'_1) = 1$, $dH(T'_1, T_2) = k$. Από επαγωγική υπόθεση $|T'_1\backslash T_2| = k$, άρα $|T_1\backslash T_2| = k – 1$ ή $|T_1\backslash T_2| = k + 1$, Αν $|T_1\backslash T_2| = k – 1$ τότε $dH(T_1, T_2) = k – 1$, άτοπο. Για να υπολογίσουμε το συντομότερο μονοπάτι θα υπολογίσουμε το σύνολο των ακμών $T_2\backslash T_1$. Για κάθε $e \in T_2 \setminus T_1$ προσθέτουμε την e στο υπάρχον δέντρο. Αν $T_2 \setminus T_1 = k$ τότε σε k βήματα θα έχουμε μετατρέψει το T_1 σε T_2 και έτσι έχουμε βρει μονοπάτι στο H μήκους k, το οποίο είναι το συντομότερο. Η συνολική πολυπλοκότητα για το ερώτημα αυτό είναι $O(k \cdot |V|)$.

C) Για αυτό το ερώτημα θα δημιουργήσουμε το MST με τον αλγόριθμο του Kruskal σε O(nlogm). Αφού το δημιουργήσουμε, για κάθε ακμή που δεν περιέχεται στο MST κάνουμε τα εξής βήματα:
   - Προσθέτουμε την ακμή στο MST και δημιουργείται ένας κύκλος.
   - Αφαιρούμε την ακμή με το μεγαλύτερο βάρος που ανήκει στον κύκλο.

   Και έτσι προκύπτει το ζητούμενο MST για κάθε ακμή, ακόμη και αν αυτή δεν περιέχεται στο αρχικό MST. Έχοντας έτοιμο το MST που περιέχει την e={u,v} αρκεί να κάνουμε ένα DFS από το u στο v κρατώντας το βάρος της βαρύτερης ακμής σε αυτό το μονοπάτι. Η πολυπλοκότητα για να το κάνουμε αυτό για κάθε ακμή είναι Θ(mn).